SECURE 64™
Software Corporation

ITANIUM® SOLUTIONS
ALLIANCE

Itanium® Architecture
Rationale & Perspective

Bill Worley
Chief Technical Officer
24 April 2006

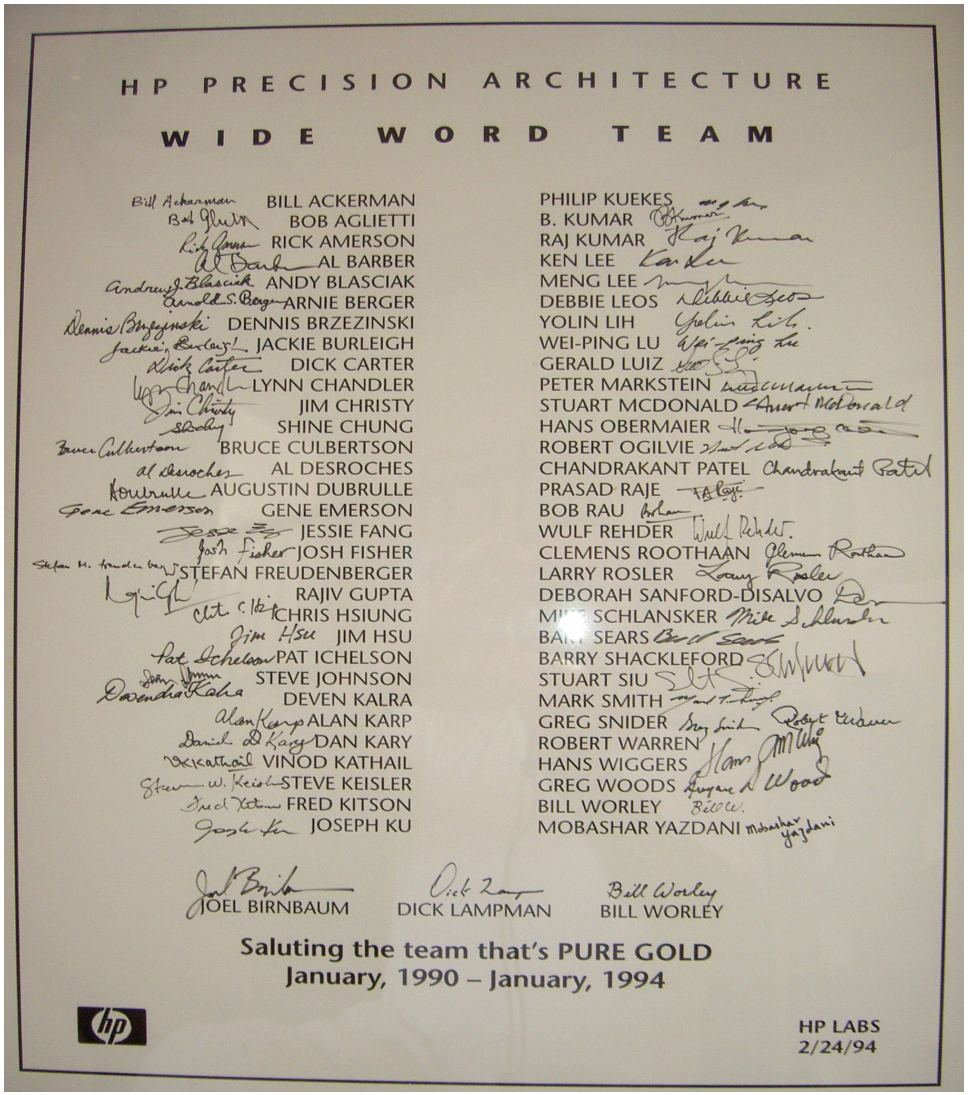
Introduction

- Two of my favorite subjects:
  - Itanium® Architecture, Underlying Science & Prospects
  - Secure systems
- HP Labs PA-WW Effort
  - Conducted Jan 1990 – Jan 1994
  - Rationale, Principles, Tech Basis of HP/Intel Partnership
- Secure64 Focus:
  - Member of Itanium Solutions Alliance
  - Itanium 2®-based, Inherently Secure, Self-Protecting Systems for the Enterprise
  - Fully employ Itanium® Architecture Security Capabilities

Topics

- Initial HP Labs Work – the PA-WW Program
- Itanium® Processor Performance Capabilities
- Itanium® Architecture Security Capabilities
- Security Landscape
- Secure64 Itanium 2®-based Secure System S/W Architecture
- Conclusions

HP Labs Work Jan 90 – Jan 94



- Chartered - Jan 1990
- Goal - Basic Advance
  - -Processor Architecture
- Process
  - Multi-Disciplinary Team
  - Investigations, Papers
  - Reference Chip Design
- Basic Findings ~1Q91
  - O^3S^2 Hard Limits
    - IPC, Complexity
    - Frequency
  - O^3S^2 Hogs Si Area
  - Compiler Must Schedule
    - All Cases
- First HP/Intel Mtg - 12/93
  - Down from Bill's & Dave's offices

Itanium® Processor Performance Capabilities

Results of fundamental architecture principles:

- Surmount CISC, RISC ILP limitations
- Higher parallelism
- Shorter Pipelines
- Smaller silicon area
- Lower branch densities
- Early memory fetch
- Cache occupancy control

Itanium® Architecture Security Capabilities

Not as widely discussed as other features

- Four static hardware privilege levels, PL0-PL3
- Virtual page protections
  - 12 page sizes, from 4K bytes to 4G bytes
  - Page access rights: R/W/X as function of H/W privilege level
  - Page protection keys: 16.7M IDs; R/W/X disable bits
- Interruptions retain page mappings, protections
- Register Save Engine
  - save/restore registers in protected pages
- Montecito - enhanced H/W authentication of firmware
- Expect future architecture enhancements

Security Landscape

- Forces Driving Modern operating systems
  - Portability
  - Race to Full Generality and “Cool” Functionality
- Implications
  - Least common denominator hardware protections
    - 40 year old, static, hardware protection model
    - I/O drivers at PL0, no H/W I/O addressing protection
  - Massive complexity, massive code at PL0
  - Dynamically loaded executable code
  - Multiple static & dynamic points of attack

Security Landscape (cont'd)

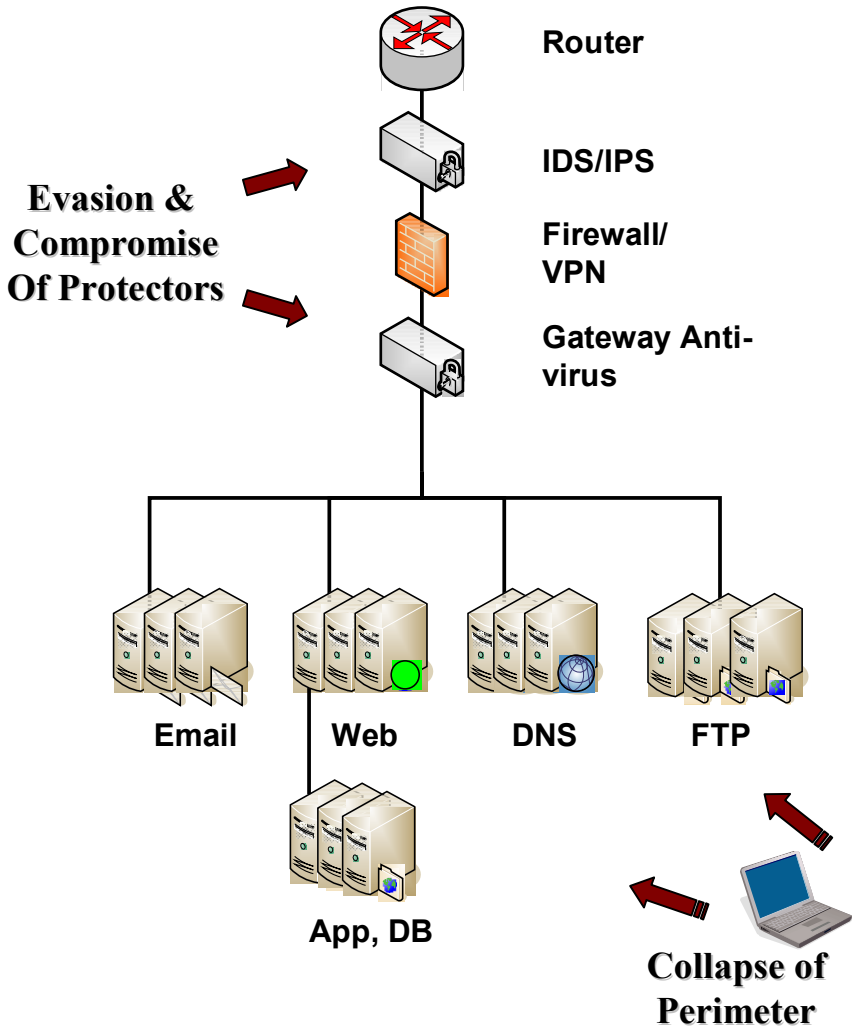
"Complexity is the worst enemy of security, and it almost always comes in the form of features or options"

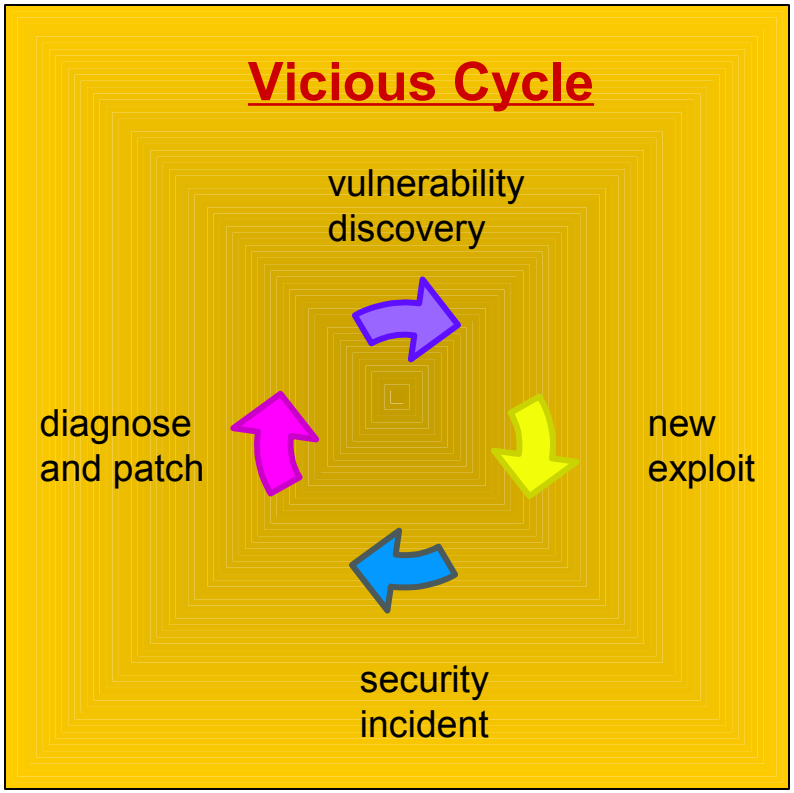
-- Niels Ferguson, Bruce Schneier, "Practical Cryptography," Wiley, 2003

Root problems for existing systems

- Too many complex interactions and options
- Too much code running at PL0
- No structural basis for strong security properties

Security Landscape (cont'd)





Band-Aids & Bodyguards

ITANIUM® SOLUTIONS ALLIANCE

SECURE64™ Software Corporation

Security Landscape (fin)

Are Band-Aids and Bodyguards working?

- RSA Conference news reports, Feb 2006
  - Panel: lots of good progress, but cyber security "not keeping pace with threats"
  - Microsoft, Sun, Cisco executives Agree:

    "It's time to build security into hardware and software from the ground up and stop trying to fix problems after the fact." – Channel Insider 2/15/06
- Microsoft, eWEEK, 4 April 2006
  - "When you are dealing with rootkits and some advanced spyware programs, the only solution is to rebuild from scratch..."
- Secure64 propositions:
  - Complex modern OSs can't be trusted to control H/W securely
  - Security should and can be designed in from the ground up
  - Only way to break the "vicious cycle"
  - Itanium® architecture uniquely enables such a design
  - Proof of concept system now in customer sites

Secure64 System S/W Architecture

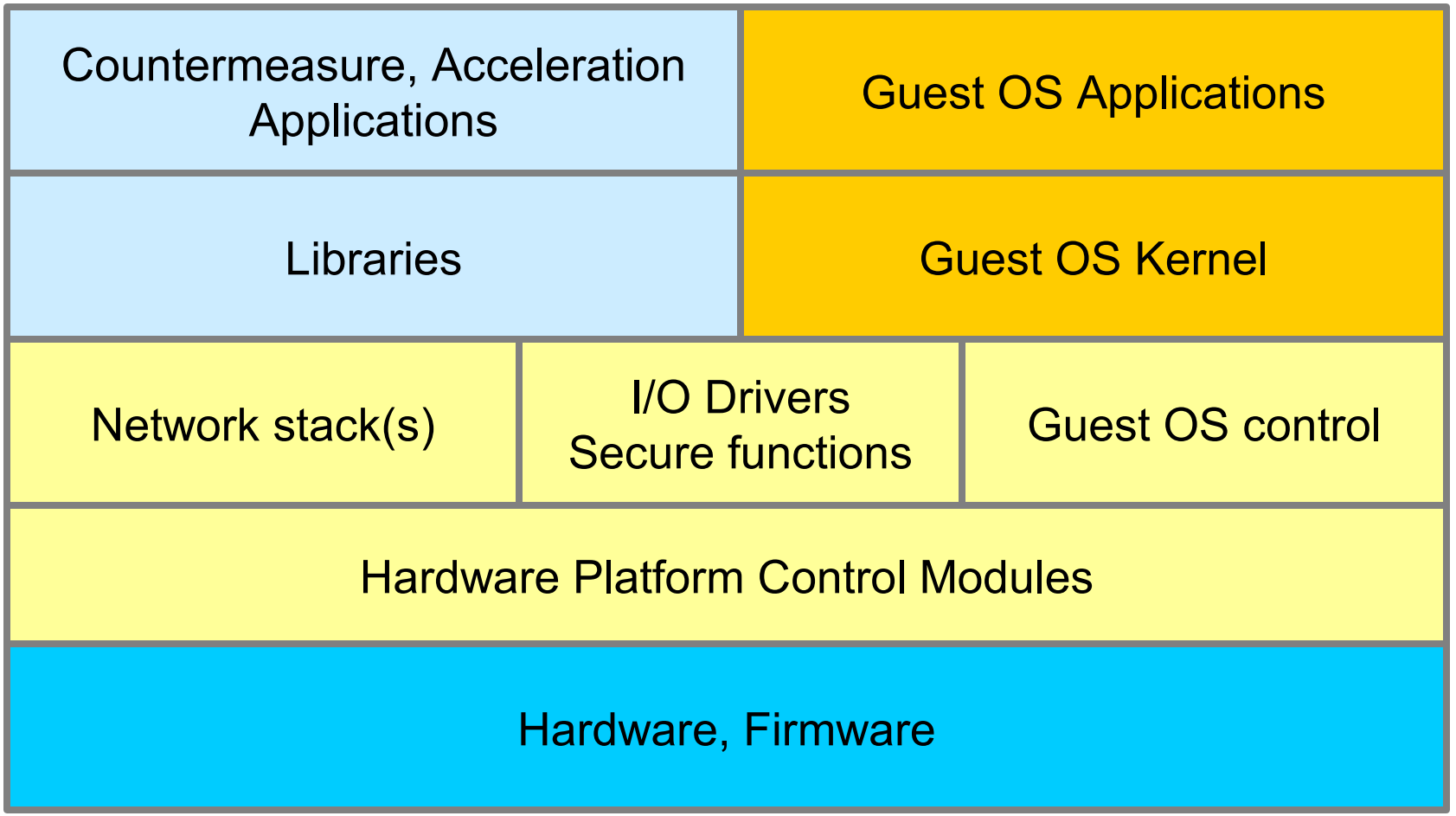
Provides two categories of S/W functionality

- Integrated Countermeasure/Acceleration Functions
- Host for Guest Operating Systems and their applications
- Functionality based on Hardware Platform Control System (HPCS)
- Inherently Secure System – specific properties
- Self-Protecting System – no bodyguards

Novel use of Itanium® architecture

- Root trust, Pervasive trust,
- Absolutely No Code Injection
- Minimum Code, no I/O drivers at PL0
- Memory compartmentalization
- Protected paths – Authenticated calls
- Protected RSE and software stacks
- Simple + light-weight + low overhead = very high performance

Basic HPCS Structure



Root Trust, Pervasive Trust

What

- Root Trust - Assurance of integrity of:
  - Firmware code images and initial data structures
  - Executable system code images and initial data structures
- Pervasive Trust – same for application code images & structures

How

- Montecito - hardware authenticates initial firmware
- PAL/SAL/EFI – could extend chain of trust to system loader
- System loader – extend chain of trust to system elements
- System – extend chain of trust to applications & guest O/S's

Guest operating systems

- Continuous monitoring of integrity
- Recovery from detectable guest OS corruption

Memory Compartmentalization

What

- Granular, static & dynamic H/W protection of memory areas
  - Control: which code accesses which data at what times
- Orthogonal to hierarchical H/W privilege levels
- Full control of R/W/X privileges, in all combinations
- Differing R/W/X privileges for different types of access/agents
- Can block all accesses, from any hardware privilege level

How

- Use all sizes of virtual pages, each tagged by protection ID
- Pin all code image page TLBs, largest data page TLBs
- Compartments built from tagged virtual pages
- Two classes of protection key management
  - Explicit request to open/close, authorized by static capability
  - Invoked/Revoked as function of authenticated call/return

Minimum PL0 Code

- No defense whatsoever from malice in PL0 code
- Today's systems: massive amounts of complex PL0 code
- PL0 code must be:
  - Mechanisms only
  - Minimized
  - Simple
  - Trusted by being published for expert critique and review
- Hardware system functions – calls to PL0 platform controls
- Platform Control calls – always authenticated

Protected Paths – Authenticated Calls

What

- Guaranteed mutually authenticated interaction path
  - User and system
  - Among system elements
- No impersonation or spoofing of a component in a path
  - e.g. Platform control calls only from authorized callers

How

- Authenticated Calls
  - By loader –call offsets in code, execute access only
    - Relies on code protection principles
  - Authenticated call mechanism – validate dynamic path

RSE and Software Stacks

Two stacks in Itanium®-based system

- Register Save Engine (RSE) stack
- Software stack

- RSE stack
  - Register values, call return addresses
  - Backed by memory inaccessible to application
  - Loader assures no RSE tampering
  - R/W access only
- Software stack
  - Protection key unique to execution thread of control
  - R/W access only
  - R access only when calling system

I/O Drivers

Principles

- I/O virtual address protection – in future H/W
- For now: must protect by software interfaces
  - Trade off: security vs. open source I/O drivers
- I/O drivers never execute at PL0
- Memory map addressability to I/O adapters
  - R only to I/O driver, R/W for PCM's
- Virtual-to-Physical address translations by PCM's
  - R only to driver, R/W for PCM's
- Key protected epc page for I/O driver to PCM functions

Conclusions

- Itanium® architecture uniquely enables an inherently secure HPCS
  - Architected H/W protections critical
  - Competing processors don't match these H/W protections
- Secure64's Self-Protecting System architecture
  - Uses Itanium® architecture H/W protections fully
  - Provides only inherently secure base for guest OS's
  - Elevates trust for guest OS's and their applications
  - Eliminates need for local bodyguard systems
  - Provides substantial customer savings
- Itanium®-based systems will be the winners for secure virtualization